Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Energy Efficiency using Adaptive Hello Scheme for MANET

Shivi Sharma*, Sonia Jangra
Department of Computer Science and Engineering,
L.R.I.E.T (Solan) H.P.T.U., Himachal Pradesh, India

Abstract— Mobile Ad- hoc Network is an emerging area of research. Most current work is centralized with different issues. MANET is temporarily constructed network with no infrastructure .It is based on self-organizing and rapidly deployed network. The special feature of MANET bring this technology great opportunity together with several challenges

A new approach, called Adaptive Hello Messaging scheme is proposed to solve the problems related to battery consumption and network overhead. The Hello Messaging Scheme aims to reduce unnecessary hello messages while neighbour discovery and also to establish a reliable connection between the source node to the destination node. This is one of the important issues that significantly affect the performance of the MANETs. The data traffic and group mobility models are also carefully chosen and configured. The behaviours of the routing protocols are tested based on the influence of node density, throughput, Hello packet overhead and energy consumed and measured using various performance metrics. The main objective is to reduce energy consumption of the nodes. In this paper we have use BFO Optimization technique to save energy in MANET, we have use three objective functions taking as an input of bfoviz time, energy and trust value and successfully overcome the energy constraint in AODV and DYMO protocol.

Keywords— MANET, Wireless Network, Ad hoc Network

I. INTRODUCTION

MANET is usually a self-organizing and self-configuring "multi-hop" network which does not require any fixed infrastructure. In such network, all nodes are dynamically and arbitrarily located, and are required to relay packets for other nodes in order to deliver data across the network.

Ad Hoc" is a Latin phrase which means "for this purpose". Individual nodes comprising the network are created dynamically and maintained. The network does not depend on a pre-existing infrastructure, e.g. routers in wired networks and therefore it is Ad-hoc.

Ad-hoc networks can be classified into three categories depending on their applications:

Mobile Ad-hoc Networks (MANETs), Wireless Mesh Networks (WMNs) and Wireless Sensor Networks (WSN).

A MANET is an autonomous collection of mobile nodes. [4] These nodes are struggling to cope with the normal effect of radio communication channels, multi-user interference, multi-path fading and shadowing etc.

Mobile Ad-hoc Networks are one of the fastest emerging network technologies. It is an unstructured network in which nodes are mobile and autonomous. Nodes act as hosts as well as routers.

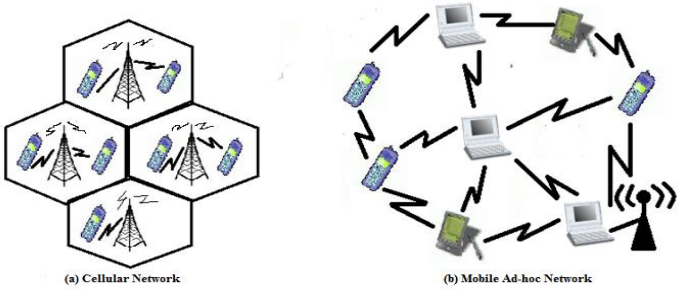


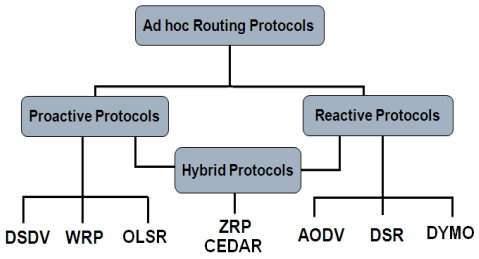
Fig. 1 Example of mobile ad-hoc network

II. ROUTING IN MANET

One of the major issues that affects the performance of an ad-hoc network is the way routing is implemented in a network. Generally, routing is the process of discovery, selecting, and maintaining paths from a source node to destination node deliver data packets. This is a main challenge in MANET, because the MANET possesses dynamic and random characteristics. Nodes move in an arbitrarily manner and at changing speed, often resulting in connectivity problems. The high mobility and the arbitrarily movement of nodes in MANET causes links between hosts to break frequently. There are many solutions for detecting link failure before packet is sent to a node. Probability of link failure of detection is calculated for this purpose. [8]

III. MANET ROUTING PROTOCOLS

The routing of traffic between nodes is performed by a MANET routing protocol. MANET routing protocols can be divided into two categories. In table driven/ proactive routing protocols, nodes periodically exchange routing information and attempt to keep up-to-date routing information. In on-demand/reactive routing protocols, nodes only try to find a route to a destination when it is actually needed for communication. In the following sections, we first describe the two categories of MANET routing protocols in more details. We then list the challenges faced by MANET routing protocols. [8]



IV. ON DEMAND ROUTING PROTOCOLS/REACTIVE PROTOCOLS

On-demand routing protocols only maintain routes that are actually used. On-demand protocols use two different operations to find and maintain routes: the route discovery process operation and the route maintenance operation. When a node wishes to communicate with some other node it tries to find a route to that node i.e. routing information is acquired on-demand. This is the route discovery operation. Route maintenance is the process of responding to changes in topology that happens after a route has initially been created. Examples of on-demand protocols are DSR, AODV and DYMO.

TABLE DRIVEN ROUTING PROTOCOLS/PROACTIVE ROUTING PROTOCOLS

Proactive routing protocols maintain routing information continuously. Typically, a node has a table containing information on how to reach every other node (or some subset hereof) and the algorithm tries to keep this table up-to-date. Changes in network topology are propagated throughout the network. Examples of proactive protocols are the Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), Highly Dynamic Destination-Sequenced Distance-Vector Routing, and OLSR.

V. THE AODV ROUTING PROTOCOL

The Ad-hoc On-Demand Distance Vector (AODV) routing protocol is a reactive protocol. As is the case with all reactive ad-hoc routing protocols; AODV consists of two protocol operations: [5]

A. Route discovery
B. Route maintenance

A. Route Discovery

Route discovery is the process of creating a route to a destination when a node lacks a route to it. When a node S wishes to communicate with a node T it initiates a Route Request (RREQ) message including the last known sequence number for T and a unique RREQ id that each node maintains and increments upon the sending of an RREQ. The message is flooded throughout the network in a controlled manner i.e. a node only forwards an RREQ if it has not done so before; the RREQ id is used to detect duplicates. Each node forwarding the RREQ creates a reverse route for itself back to S using the address of the previous hop as the next hop entry for the node originating the RREQ. When the RREQ reaches a node with a route to T (possibly T itself) a Route Reply (RREP), containing the number of hops to T and the sequence number for that route, is sent back along the reverse path. An intermediate node must only reply if it has a fresh route, i.e., the sequence number for T is greater than or equal to the destination sequence number of the RREQ. Since replies are sent on the reverse path, AODV do not support asymmetric links. Each node receiving this RREP creates a forward route to T in its routing table, and adds the node that transmitted the RREP in precursor list for this entry. The precursor list is a list of nodes that might use this node as next hop towards a destination. Route discovery is illustrated in figure 1.3, where node 2 wants to communicate with node 9 and floods an RREQ message in the network. Node 9 replies with an RREP. Intermediate nodes learn routes to both source and destination nodes via the RREQ and RREP packets.

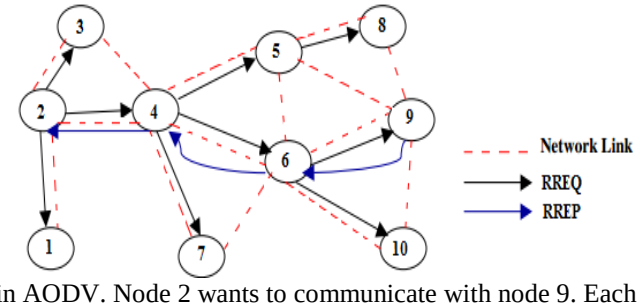


Figure. 3 Route discovery in AODV. Node 2 wants to communicate with node 9. Each node forwarding the RREQ creates a reverse route to node 2 used when sending back the RREP.

If an intermediate node has a route to a requested destination and sends back an RREP, it must discard the RREQ. Furthermore, it may send a gratuitous RREP to the destination node containing address and sequence number for the node originating the RREQ. Gratuitous RREPs are sent to alleviate any route discovery initiated by the destination node. It might not have received any RREQs and has not learned a route to the originator of the RREQ.

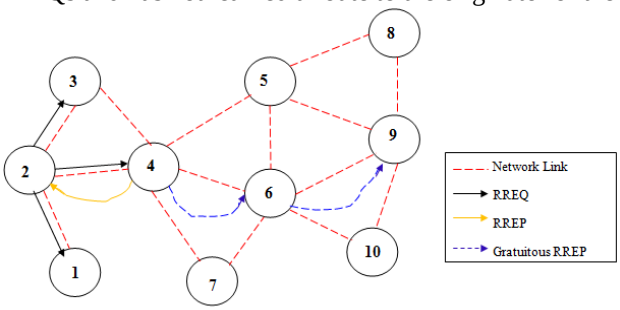


Figure.4 Generation of an RREP by an intermediate node. Node 4 has a route to node 9 and sends an RREP to node 2 and a gratuitous RREP to node 9.

B. Route Maintenance

Route maintenance is the process of responding to changes in topology. To maintain paths, nodes continuously try to detect link failures (when a neighbour goes out of range, the node itself moves, or some other event limiting the communication on the link). Nodes listen to RREQ and RREP messages to do this. Furthermore, each node promises to send a message every n seconds. If no RREQ or RREP is sent during that period, a Hello message is sent to indicate that the node is still present. Alternately, a link layer mechanism can be used to detect link failures. Beside the observation of a link failure, a node must also respond when it receives a data packet it does not have a route for. When a node detects a link break or it receives a data packet it does not have a route for, it creates and sends a Route Error (RERR) packet to inform other nodes about the error. The RERR contains a list of the unreachable destinations. If a link break occurs, the node adds the unreachable neighbour to the list. If a node receives a packet it does not have a route for, the node adds the unreachable destination to the list. In both cases, all entries in the routing table that make use of the route through the unreachable destination are added to the list. The list is pruned, as destinations with empty precursor lists, i.e., destinations that no neighbours currently make use of, are removed. The RERR message is either unicasted (in case of a single recipient) or broadcasted to all neighbours having a route to the destinations in the generated list. This specific set of neighbours is obtained from the precursor lists of the routing table entries for the included destinations in the RERR list. When a node receives an RERR, it compares the destinations found in the RERR with the local routing table and any entries that have the transmitter of the RERR as the next hop, remains in the list of unreachable nodes. The RERR is then either broadcasted or unicasted as described above. The intention is to inform all nodes using a link when a failure occurs. For example, in figure 1.5, a link between node 6 and node 9 has broken and node 6 receives a data packet for node 9. Node 6 generates a RERR message, which is propagated backwards toward node 2.

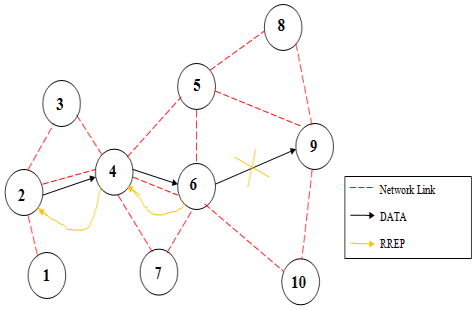


Figure.5 Generation of RERR messages. The link between node 6 and node 9 has broken, and node 6 generates an RERR.

To find a new route, the source node can initiate a route discovery for the unreachable destination, or the node upstream of the break may locally try to repair the route, in either case by sending an RREQ with the sequence number for the destination increased by one.

VI. THE DYMO ROUTING PROTOCOL [10]

The Dynamic MANET On-demand DYMO routing protocol is a newly proposed protocol currently defined in an IETF Internet-Draft in its sixth revision and is still work in progress. DYMO is a successor of the AODV routing protocol and is the current engineering focus for reactive routing in the IETF MANET working group. It operates similarly to AODV. DYMO does not add extra features or extend the AODV protocol, but rather simplifies it, while retaining the basic mode of operation. Routes are discovered on-demand when a node needs to send a packet to a destination currently not in its routing table. A route request message is flooded in the network using broadcast and if the packet reaches its destination, a reply message is sent back containing the discovered, accumulated path. Each node maintains a routing table with information about nodes. As is the case with all reactive ad-hoc routing protocols, DYMO consists of two protocol operations:

A. Route discovery
B. Route maintenance

A. Route Discovery

Route discovery is the process of creating a route to a destination when a node needs a route to it. When a node S wishes to communicate with a node T, it initiates a Route Request (RREQ) message. The RREQ message and the Route Reply (RREP) message, which we describe later in this section, are collectively known as Routing Messages (RM) because they are used to distribute routing information. The sequence number maintained by the node is incremented before it is added to the RREQ. We illustrate the route discovery process using figure 1.6 as an example. In the figure 1.6, node 2 wants to communicate with node 9 and thus, node 2 is S, the source, and node 9 is T, the target destination. In the RREQ message, the node 2 includes its own address and its sequence number, which is incremented before it is added to the RREQ. It can also include prefix value and gateway information if the node is an Internet gateway capable of forwarding packets to and from the Internet. Finally, a hop count for the originator is added with the value 1. Then information about the target destination 9 is added. The most important part is the address of the target. If the originating node knows a sequence number and hop count for the target, these values are also included. To sum up, the RREQ so far contains information about node 2 that originated the RREQ and node 9, the target destination. The message is flooded using broadcast, in a controlled manner, throughout the network, i.e., a node only forwards an RREQ if it has not done so before. The sequence number is used to detect this. Each node forwarding an RREQ may append its own address, sequence number, prefix, and gateway information to the RREP. Upon sending the RREQ, the originating node will await the reception of an RREP message from the target. If no RREP is received within RREQ WAIT TIME, the node may again try to discover a route by issuing another RREQ. RREQ WAIT TIME is a constant defined in the DYMO specification and the default value is 1000 milliseconds. In figure 1.6, the nodes 4 and 6 append information to the RREQ when they propagate the RREQ from node 2. When a node receives an RREQ, it processes the addresses and associated information found in the message.

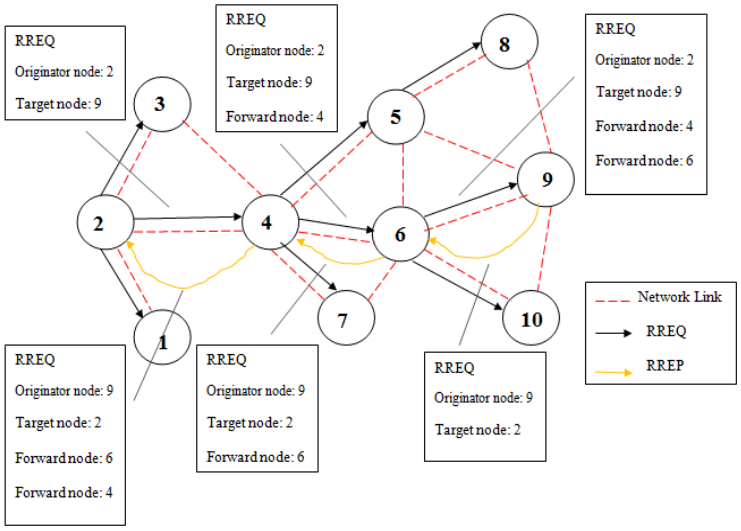


Figure.6 The DYMO route discovery process. Node 2 wants to communicate with node 9. Each node forwarding the RREQ creates a reverse route to 2 used when sending back the RREP. When sending back the RREP, nodes on the reverse route create routes to node 9.

The information for a node 1 is compared with the corresponding entry in the routing table of the node, if one exists. The information about the originator found in the RREQ is processed first, but subsequent entries are processed the same way:

- If the routing table does not contain an entry for the originator, one is created. The next hop entry is the address of the node from which the RREQ was received. Likewise, the next hop interface is the interface on which the RREQ was received.
- If an entry exists, the sequence number and hop count found in the RREQ is compared to the sequence number route and hop count in the table entry to check if the information in the RREQ is stale or should be disregarded.
- If an entry exists and is not stale or disregarded, the entry is updated with the information found in the RREQ. If the originator entry in the RREQ is found to be stale or disregarded, the RREQ is dropped. For other nodes, the information is removed from the RREQ.
- If an RREQ is not dropped, each node processing the RREQ can create reverse routes to all the nodes for which addresses are accumulated in the RREQ.

Upon adding an entry for I to its route table entry, the node processing the RREQ increments the hop count value for I found in the RREQ to correctly reflect the number of hops the RREQ has travelled since the node I added its own information to the RREQ. When the RREQ reaches the destination node 9, it processes the packet similar to nodes that have forwarded the packet, and uses the information accumulated in the RREQ to add route table entries. Specifically, an

entry for the node 2 that originated the RREQ is installed. An RREP message is then created as a response to the RREQ, containing information about node 9, i.e., address, sequence number, prefix, and gateway information, and the RREP message is sent back along the reverse path using unicast. Since replies are sent on the reverse path, DYMO does not support asymmetric links. The packet processing done by nodes forwarding the RREP is identical to the processing that nodes forwarding an RREQ perform, i.e., the information found in the RREP can be used to create forward routes to nodes that have added their address block to the RREP. We shortly summarize the route discovery process depicted in figure 1.6 Node 2 wants to communicate with node 9 and floods an RREQ message in the network. As can be seen in the figure, when node 2 begins route discovery, the RREQ initially contains the address of the originator and target destination. When node 4 receives the RREQ, it installs a route to node 2. After node 4 has forwarded the RREQ, it has added its own address to the RREQ, which means it now contains three addresses. Identical processing occurs at node 6 and it installs a route to node 2 with a hop count of 2 and node 4 as the next hop node. When node 9 receives the RREQ, it contains four addresses and has travelled three hops. Node 9 processes the RREQ and install routes using the accumulated information and as it is the target of the RREQ, it furthermore creates an RREP as a response. The RREP is sent back along the reverse route. Similar to RREQ dissemination, every node forwarding the RREP adds its own address to the RREP and installs routes to node 9.

B. Route Maintenance [10]

Route maintenance is the process of responding to changes in topology that happens after a route has initially been created. To maintain paths, nodes continuously monitor the active links and update the Valid Timeout field of entries in its routing table when receiving and sending data packets. If a node receives a data packet for a destination it does not have a valid route for, it must respond with a Route Error (RERR) message. When creating the RERR message, the node makes a list containing the address and sequence number of the unreachable node. In addition, the node adds all entries in the routing table that is dependent on the unreachable destination as next hop entry. The purpose is to notify about additional routes that are no longer available. The node sends the list in the RERR packet. The RERR message is broadcasted. The dissemination process is illustrated in figure 1.6. A link between node 6 and node 9 breaks and node 6 receives a data packet for node 9. When we say a link is broken, it could just be that the time stamp in the route table entry for a node timed out and the entry has become invalid. Node 6 generates an RERR message, which is propagated backwards towards node 2.

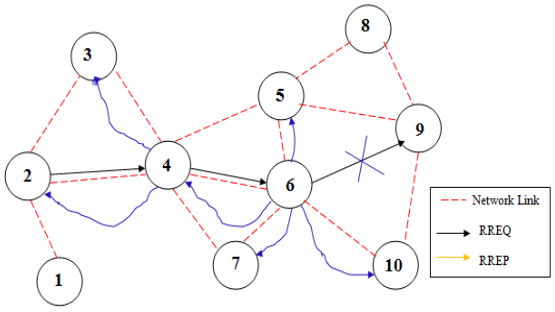


Figure.7 Generation and dissemination of RERR messages. The link between nodes 6 and 9 breaks, and node 6 generates an RERR. Only nodes having a route table entry for node 9 propagate the RERR message further.

When a node receives an RERR, it compares the list of nodes contained in the RERR to the corresponding entries in its routing table. If a route table entry for a node from the RERR exists, it is invalidated if the next hop node is the same as the node the RERR was received from and the sequence number of the entry is greater than or equal to the sequence number found in the RERR. If a route table entry is not invalidated, the corresponding entry in the list of unreachable nodes from the RERR must be removed. If no entries remain, the node does not propagate this RERR further. Otherwise, the RERR is broadcasted further. The sequence number check mentioned is performed to only invalidate fresh routes and to prevent propagating old information. The intention of the RERR distribution is to inform all nodes that may be using a link, when a failure occurs. RERR propagation is guaranteed to terminate as a node only forwards an RERR message once. In figure 1.7, when the RERR is broadcasted, additional nodes beside node 4 and 2 will receive the message, for example, the nodes 5, 7, and 10. As none of these use node 6 as a next hop towards node 9, they all drop the RERR after processing the message. In addition to acting upon receiving a packet to a destination without a valid route table entry, nodes must continuously try to detect link failures to maintain active links. Link failures occur, for example, when a neighbour goes out of range, the node itself moves, or some other event limiting the communication on the link. The mechanisms used by a node to monitor active links can be Hello messages, link layer feedback, neighbour discovery, or route timeouts. Hello messages are packets that are periodically broadcasted with the intent of detecting the presence or disappearance of neighbours. However, the fourth revision DYMO specification draft does not specify the use or packet layout of Hello messages. As of the fifth revision of the DYMO specification draft, the use of Hello messages and the unspecified neighbour discovery have been updated to suggest the use of neighbourhood discovery as specified in the MANET Neighbourhood Discovery Protocol (NHDP). If a broken link is detected, the node may disseminate an RERR to notify other nodes about the broken link. The process is identical to the one described above. Finally, when a node receives an RERR for a destination, to rediscover a route, the node can initiate a route discovery for the unreachable destination by sending an RREQ message.

VII. PROPOSED METHODOLOGY

Following are the various steps required to successfully simulate the proposed algorithm.

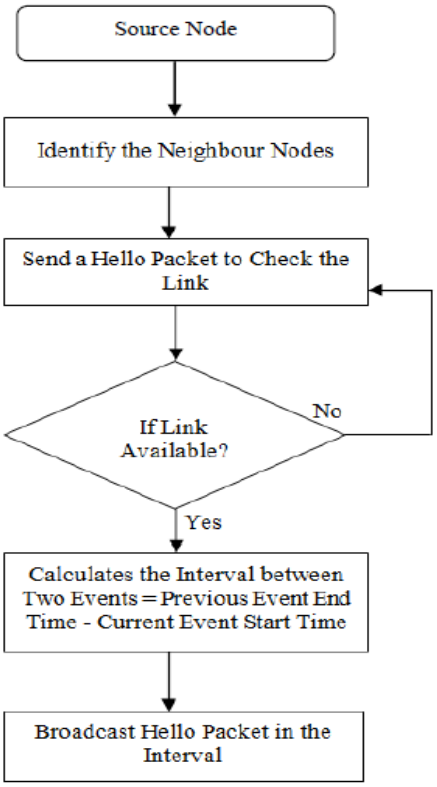


Figure.8 Flowchart of the proposed technique

Step 1: First of all initialize ad-hoc network with their respective characteristics like moving range, maximum dimensions, number of nodes etc.

Step 2: Define cluster heads having multi-radio and multi-channel facility.

Step 3: Sender(s) will be initiated to multicast its data to defined nodes.

Step 4: Sender will hand over its data to nearest cluster head using Euclidian distance.

Step 5: Cluster head will multicast data to available cluster heads depends upon the ACO based shortest path.

Step 6: Evaluate energy dissipation as well as other QoS features, and move to step 3

VIII. RESULTS AND DISCUSSIONS

A. EXECUTION TIME

In this research, a new approach, called **Adaptive Hello Messaging** scheme is proposed to solve the problems related to battery consumption and network overhead. The Hello Messaging Scheme aims to reduce unnecessary hello messages while neighbour discovery and also to establish a reliable connection between the source node to the destination node. This is one of the important issue that significantly affect the performance of the MANETs. The data traffic and group mobility models are also carefully chosen and configured. The behaviours of the routing protocols are tested based on the influence of node density, throughput, Hello packet overhead and energy consumed and measured using various performance metrics. The main objective was to reduce energy consumption by the nodes. Exponential traffic model is chosen for the research. Average event interval between the nodes is calculated by its formula. By this Hello message interval is adjusted on various number of flows, number of events and number of nodes. It was observed in the results that at within some time interval maximum number of nodes are not involved in the communication. This means that there is no need of sending Hello messages at that particular time. After applying adaptive scheme results of both protocols i.e. AODV and DYMO are compared with adaptive scheme protocols i.e. AODV-AH, DYMO-AH. Also comparison between AODV-AH and DYMO-AH is done. DYMO-AH consumes less energy as compared to AODV-AH. In second part of thesis optimization using Bacterial foraging optimization Algorithm (BFOA) is done on protocols with adaptive scheme. In BFO algorithm bacterium named E.coli's movement is calculated. It has two movements i.e. swim and tumble. It's for optimization are chemotaxis, reproduction, elimination and dispersal. In BFO parameters values of trust value of nodes, energy and delay values are fed. Maximization is performed, if optimized value occurs then optimization is done otherwise it is fed back to initialized values. Optimized results are named AODV-AHBFO and DYMO-AHBFO. Comparison with adaptive scheme protocols and both optimized protocols are also done. At the end we have concluded that battery consumption can be reduced much more using global optimization algorithms like BFOA.
Proposed an adaptive Hello interval to reduce battery drain through practical suppression of unnecessary Hello messaging. Based on the event interval of a node, the Hello interval can be enlarged without reduced detectability of a broken link, which decreases network overhead and hidden energy consumption.

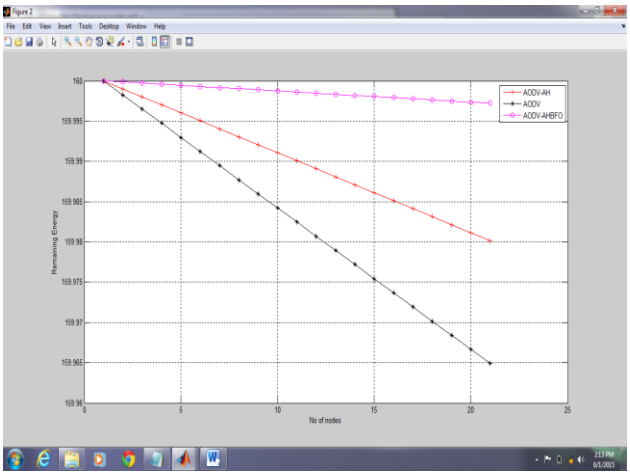
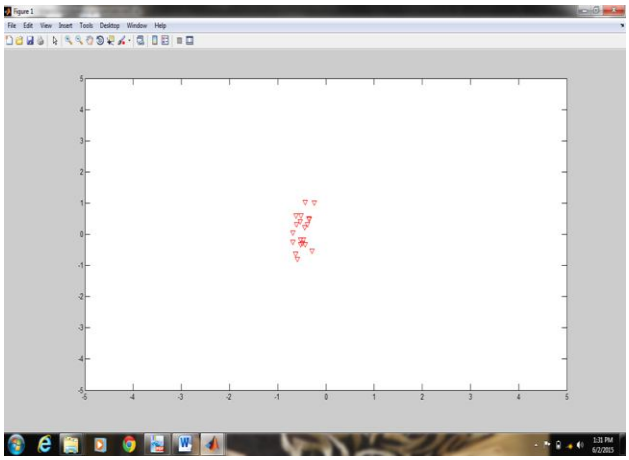


Fig 1.7 Energy consumption with variable nodes of AODV



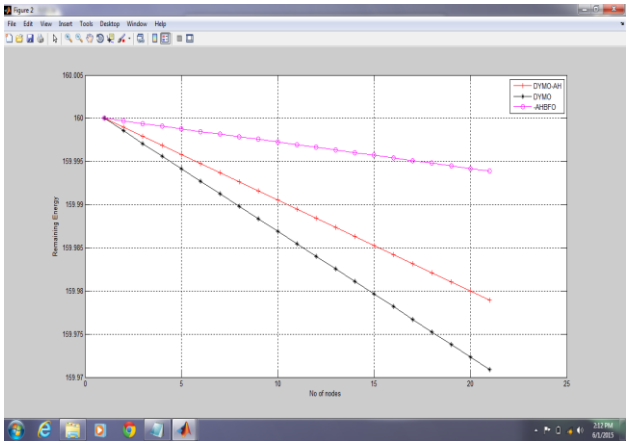


Fig 1.8 Energy consumption with variable nodes of DYMO using bfo

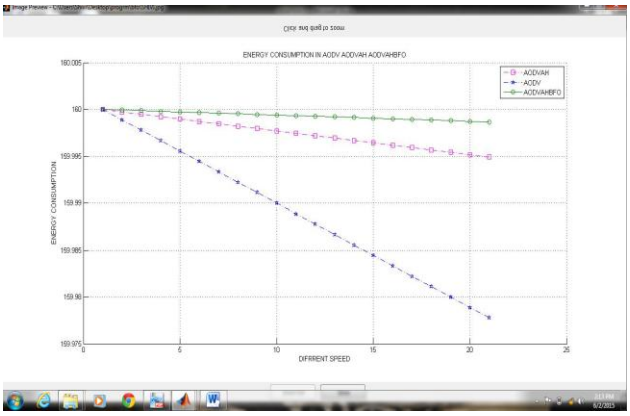


Fig 1.9 Energy consumption in AODV AODVAH AODVAHBFO

IX. CONCLUSION

This paper provides a performance analysis of two different mobile ad-hoc routing protocols, namely, AODV and DYMO using an MATLAB. In this research, a new approach, called **Adaptive Hello Messaging** scheme is proposed to solve the problems related to battery consumption and network overhead. The Hello Messaging Scheme aims to reduce unnecessary hello messages while neighbour discovery and also to establish a reliable connection between the source node to the destination node. This is one of the important issue that significantly affect the performance of the MANETs. The data traffic and group mobility models are also carefully chosen and configured. The behaviours of the routing protocols are tested based on the influence of node density, throughput, Hello packet overhead and energy consumed and measured using various performance metrics. The main objective was to reduce energy consumption by the nodes. Exponential traffic model is chosen for the research. Average event interval between the nodes is calculated by its formula. By this Hello message interval is adjusted on various number of flows, number of events and number of nodes. It was observed in the results that at within some time interval maximum number of nodes are not involved in the communication. This means that there is no need of sending Hello messages at that particular time. After applying adaptive scheme results of both protocols i.e. AODV and DYMO are compared with adaptive scheme protocols i.e. AODV-AH, DYMO-AH. Also comparison between AODV-AH and DYMO-AH is done. DYMO-AH consumes less energy as compared to AODV-AH. In second part of thesis optimization using Bacterial foraging optimization Algorithm (BFOA) is done on protocols with adaptive scheme. In BFO algorithm bacterium named E.coli's movement is calculated. It has two movements i.e. swim and tumble. It's for optimization are chemotaxis, reproduction, elimination and dispersal. In BFO parameters values of trust value of nodes, energy and delay values are fed. Maximization is performed, if optimized value occurs then optimization is done otherwise it is fed back to initialized values. Optimized results are named AODV-AHBFO and DYMO-AHBFO. Comparison with adaptive scheme protocols and both optimized protocols are also done. At the end we have concluded that battery consumption can be reduced much more using global optimization algorithms like BFOA.

X. FUTURE SCOPE

As a future work, we can deploy the proposed approach in more scenarios and large scale networks. Adaptive scheme can be applied on other protocols of MANET like DSDV, DSR, OLSR, TORA etc. We plan to extend our work to optimize the value of the hello intervals with more number of nodes. Also optimization can be done by using more different optimization techniques like Particle Swarm Optimization (PSO), Ant Colony algorithm, Bee algorithm, Genetic algorithm (GA). Also there are security problems in MANET protocols like black hole. We can work on these security problems of these protocols.

REFERENCES

[1] D.Remondo ," Tutorial of Wireless Ad hoc Network", HET- NET S,2004

[2] Mohammad IIyas, "The hand book of Ad hoc wireless Network ", CRC press LEC.

[3] S. Ali, and A. Ali, "Performance Analysis of AODV, DSR and OLSR in MANET",Masters Thesis, M.10:04, COM/School of Computing, BTH, 2010. [Online]. Availableat:http://www.bth.se/fou/cuppsats.nsf/all/252aefb4936b9db3c12576b20053b8a5/$file/Performance%20Analysis%20of%20AODV%2C%20DSR%20and%20OLSR%20in%20MANET.pdf

[4] M.K. J. Kumara and R.S. Rajesh, "Performance Analysis of MANET Routing Protocols in

[5] Different Mobility Models" IJCSNS *International Journal of Computer Science and Network 22 Security*, VOL.9 No.2, February2009.

[6] http://en.wikipedia.org/wiki/Mobile_ad_hoc_network.

[7] Arun kuamr B.R, Lokanatha C.Reddy "Mobile Ad Hoc Network: Issues, Research Trends and Experiments" IETECH journal of Communication Techniques, Vol: 2, No: 2, 057-063.

[8] Aart, Dr. S.S Tyagi, "Study of MANET: Characteristics, Challenges, Application and Security Attacks "International Journal of Advance Research in Computer Science and SoftwareEngineering, Volume 3, Issue 5, May 2013.

[9] Joroen Hoebeka, Bark ,"An overview of mobile Ah Hoc Network Application and Challenges, Ghent University – IMEC vzw, Sint Pietersnieuwstraat 41, B-9000 Ghent.

[10] Jatinder Pal Singh, Anuj Kr. Gupta, "A Review on Dynamic MANET On- Demand Routing Protocol in MANETs". International Journal of Advanced Trends in Computer Science and Engineering, Volume 2, No.2, March - April 2013.

[11] Rolf Ehrenreich thorup, Lars Kristensen, "Implementing and Evaluating the DYMO Routing Protocol". Master's Thesis, February, 2007.

[12] Ehsan Mostajeran, Rafidah Md Noor et al., "A Novel Improved Neighbour Discovery Method for an Intelligent-AODV in Mobile Ad-hoc Networks".IEEE, 978-1-4673-4992, May 2013.

